

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería informática

TRABAJO FIN DE GRADO

RECONSTRUCCIÓN VIRTUAL DE EDIFICIOS HISTÓRICOS

Tatiana Melissa Naranjo Castaño
Tutor: Sacha Gómez Moñivas

Julio 2015

RESUMEN

Este documento presenta todo el proceso de desarrollo de este proyecto, desde la motivación de la propuesta hasta las conclusiones del mismo. Se analizan y exponen las fases de análisis de requisitos, diseño, implementación y pruebas que componen el ciclo de vida del mismo.

Gracias a internet, los museos y monumentos históricos han dado el paso de extender horizontes y abrir sus puertas a todos aquellos que quieran visitarlos a través de la red. Para ello se han creado las visitas virtuales. Estas visitas suelen estar compuestas por fotos de 360º de cada estancia del edificio a recorrer. Aunque este tipo de visita cumple el propósito de ofrecer el contenido de los museos a los visitantes virtuales, la falta de interactividad de este sistema se queda corto dadas las tecnologías existentes en la actualidad.

Este Trabajo de Fin de Grado describe y presenta una visita 3D creada para todos aquellos que no puedan visitar el edificio histórico Templo de Debod y para aquellos que desean utilizarla como una herramienta didáctica. Permite tanto hacer un tour completo como recorrer de forma libre cada una de las estancias, siempre con acceso a la información relevante del monumento.

Con este fin se han utilizado herramientas de modelado 3D, herramientas de tratamiento de imagen y herramientas de desarrollo de entorno y de aplicaciones interactivas, aplicando durante todo el ciclo de vida del proyecto las técnicas de administración y gestión de proyectos de software estudiadas y practicadas durante el estudio del Grado en Ingeniería Informática.

Para complementar esta aplicación se ha creado un sistema externo para gestionar la información que se expone al visitante utilizando las conexiones a bases de datos y los accesos a servidores aprendidos durante la carrera.

PALABRAS CLAVE

Visita virtual, Templo de Debod, modelado 3D, texturizado, renderizado.

ABSTRACT

This document presents the entire process of development of this project, from the motivation of the proposal to the conclusions. It analyzes and exposes the phases of requirements analysis, design, implementation and testing of this life cycle.

Thanks to the Internet, museums and historical monuments have taken the step of expanding horizons and open their doors to all who want to visit them through the network. This has been done by creating virtual tours. These visits are usually composed of 360º photos of each room of the museum. Although this visit serves the purpose of providing the content to the virtual visitors, this system's lack of interactivity falls behind given the existing technologies today.

This Final Degree Work describes and presents a 3D tour created for those who can't visit the Temple of Debod historic building and for those who wish to use it as a teaching tool. It allows both a full tour and a free travel through each of the rooms, providing also access to the relevant information of the monument.

To achieve this, 3D modeling tools, image processing tools and interactive applications development tools have been used, while applying throughout the project life cycle management techniques and software project management techniques studied and practiced during studying the Degree in Computer Engineering.

As a complement to this application, an external system has also been created, to manage the information that is exposed to visitors using connections to databases and access to servers learned during the race.

KEYWORDS

Virtual tour, Debod Temple, 3D modeling, texturing, rendering.

AGRADECIMIENTOS

En primer lugar quiero agradecer a mi tutor, Sacha Gómez Moñivas, por el apoyo que me ha prestado en la realización del proyecto. También al Departamento de Historia Antigua, Medieval y Paleografía y Diplomática de la UAM por facilitarme los planos del Templo Egipcio de Debod y al profesor Miguel Ángel Molinero Polo, por permitirnos entrar al templo en uno de los días que estaba oficialmente cerrado al público.

Quiero agradecer a mi familia que me ha animado y apoyado en el transcurso de la carrera a seguir, pese a haberla comenzado con mal pie.

Sobre todo quiero agradecer a mi novio Alejandro Ortiz Martín por estar a mi lado siempre que lo necesitaba, animándome y apoyándome.

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN.....	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Ciclo de vida del software	2
1.4. Estructura del documento	3
2. ESTADO DEL ARTE	5
2.1. Estudio de visitas virtuales.....	5
2.1.1. Visita virtual del Templo Egipcio de Debod.....	6
2.1.2. Visita 3D del Templo Egipcio de Debod.....	7
2.1.3. Visita virtual del Museo del Romanticismo.....	8
2.1.4. Visita virtual del Museo de Antropología.....	9
2.1.5. Conclusiones del estudio.....	10
2.2. Herramientas de modelado 3D	11
2.2.1. Herramientas de diseño gráfico	11
2.2.2. Herramientas de desarrollo y de aplicaciones.....	17
3. ANÁLISIS DE REQUISITOS.....	21
3.1. Requisitos funcionales.....	21
3.1.1. Administrador	21
3.1.1. Visitante.....	21
3.2. Requisitos no funcionales	22
3.2.1. Seguridad	22
3.2.2. Interfaz	22
3.2.3. Portabilidad	23
3.2.4. Rendimiento	23
3.2.5. Usabilidad.....	23
3.2.6. Accesibilidad.....	24
4. DISEÑO DEL SOFTWARE	25

4.1.	Diseño general	25
4.2.	Diseño modular de la visita 3D	26
4.2.1.	Gestor de escenas.....	26
4.2.2.	Gestor de movimiento.....	27
4.2.3.	Gestor del menú de opciones	27
4.2.4.	Gestor del transporte	27
4.2.5.	Gestor del tour.....	28
4.2.6.	Gestor de ventanas emergentes.....	28
4.2.7.	Gestor de acceso a base de datos	28
4.3.	Diseño aplicación de administración	29
4.3.1.	ViewLogin.....	29
4.3.2.	ViewMenu.....	30
4.3.3.	ViewInstance	30
4.3.4.	RequestController.....	30
4.3.5.	ServerController	30
4.4.	Estructura de la base de datos	30
4.4.1.	Tabla language.....	30
4.4.2.	Tabla Text	31
5.	IMPLEMENTACIÓN	33
5.1.	Modelado 3D del templo.....	33
5.2.	Texturizado.....	34
5.3.	Integración del modelado 3D en Unity	36
5.4.	Implementación de la aplicación en Unity	37
5.4.1.	Gestor de escenas.....	38
5.4.2.	Gestor de movimiento.....	38
5.4.3.	Gestor del menú de opciones	39
5.4.4.	Gestor del transporte	40
5.4.5.	Gestor del tour.....	41
5.4.6.	Gestor de ventanas emergentes.....	42

5.4.7.	Gestor de acceso a base de datos	43
5.5.	Implementación de la aplicación del administrador	44
5.5.1.	Vistas	44
5.5.2.	RequestController.....	44
5.5.3.	ServerController	44
6.	PRUEBAS	47
6.1.	Pruebas de movimiento	47
6.1.1.	Prueba de movimiento de traslación	47
6.1.2.	Prueba de movimiento de rotación	47
6.2.	Pruebas de física	47
6.3.	Pruebas de conexión con la base de datos	48
6.3.1.	Prueba de aparición de textos	48
6.3.2.	Prueba de cambio de idioma	48
7.	CONCLUSIONES Y TRABAJO FUTURO.....	49
8.	REFERENCIAS	51
A.	ANEXO: ADQUISICIÓN DE MATERIAL AUDIOVISUAL	A
B.	ANEXO: CONCEPT ART.....	C
C.	ANEXO: MANUAL DE USUARIO.....	E

ÍNDICE DE FIGURAS

Ilustración 1. Modelo de ciclo de vida en cascada	3
Ilustración 2. Visita virtual Templo Egipcio de Debod.....	6
Ilustración 3. Visita 3D Templo Egipcio de Debod.....	7
Ilustración 4. Visita museo del Romanticismo	8
Ilustración 5. Visita museo del Antropología	9
Ilustración 6. Interfaz Autodesk Maya	12
Ilustración 7. Interfaz Autodesk 3D Max.....	13
Ilustración 8. Objetos básicos SketchUp.....	14
Ilustración 9. Interfaz SketchUp	15
Ilustración 10. Interfaz Blender	16
Ilustración 11. Objetos Blender	16
Ilustración 12. Second Life.....	18
Ilustración 13. Interfaz Unity.....	19
Ilustración 14. Esquema general del sistema.....	25
Ilustración 15. Esquema general de la aplicación.	26
Ilustración 16. Esquema de la aplicación del administrador.	29
Ilustración 17. Estructura de la base de datos.....	30
Ilustración 18. Alzamiento del templo en Blender.....	33
Ilustración 19. Modelo 3D del Templo Egipcio de Debod.	34
Ilustración 20. Creación de textura del alzado principal del templo en Manga Studio 5.0	35
Ilustración 21. Creación de textura en Manga Studio 5.0	35
Ilustración 22. Texturizado por UV Mapping en Blender.	35
Ilustración 23. Creación de materiales en GIMP 2.	36
Ilustración 24. Templo Egipcio de Debod integrado en Unity.	37
Ilustración 25. Esquema de flujo del menú de la aplicación.	40
Ilustración 26. Diagrama de estado del gestor de ventanas emergentes.	43
Ilustración 27. Visita al Templo Egipcio de Debod (profesor Miguel Ángel frente al Alzado Principal del templo).	A
Ilustración 28. Menú principal.	C
Ilustración 29. Selección de salas.....	C
Ilustración 30. Créditos de la aplicación.....	C
Ilustración 31. Opciones: configuración de las velocidades del desplazamiento y la rotación.....	D

ÍNDICE DE TABLAS

Tabla 1. Requisitos funcionales del administrador.	21
Tabla 2. Requisitos funcionales del visitante.	22
Tabla 3. Requisitos no funcionales de seguridad.	22
Tabla 4. Requisitos no funcionales de interfaz.	22
Tabla 5. Requisitos no funcionales de portabilidad.	23
Tabla 6. Requisitos no funcionales de rendimiento.	23
Tabla 7. Requisitos no funcionales de usabilidad.	23
Tabla 8. Requisitos no funcionales de accesibilidad.	24

GLOSARIO

Iluminación indirecta: Es un grupo de algoritmos usados en gráficos por computadora tridimensionales que tienen como objetivo añadir realismo a la modelación de la luz en escenas 3D.

Renderizar: Proceso que genera una imagen o video mediante el cálculo de iluminación indirecta partiendo de un modelo en 3D.

Visita virtual: simulación de un espacio real, esta se compone de dos o varias imágenes dimensionales unidas mediante un software especial, creando una imagen interactiva que el concurrente puede controlar mediante el ordenador, visualizándolo en 360°. Lo que permite decir que una visita virtual es lo más parecido a estar realmente en el sitio.

Mapear: Es el método que envuelve una textura alrededor de un objeto.

Modelado 3D: Es el proceso de realizar una función matemática a cualquier objeto tridimensional utilizando un software especializado.

Modelo 3D: Es el objeto resultante del modelado 3D.

Uv mapping: Es una manera de mapear texturas de tipo Imagen sobre modelos tridimensionales. Se utiliza para objetos complejos.

1. INTRODUCCIÓN

En este capítulo se exponen los motivos y objetivos por los que se ha realizado la reconstrucción del edificio histórico: Templo Egipcio de Debod, así como la estructura de la memoria.

1.1. Motivación

Cada vez hay más museos y monumentos históricos que deciden dar el salto a las visitas virtuales.

Estas visitas virtuales se ha podido crear gracias a la accesibilidad que internet da a los usuarios. Esto permite que aquellos que no podían acceder en persona al monumento, o quieran darle un uso didáctico, tengan un modo de hacerlo.

Aunque hay muchas visitas virtuales, la mayoría se hacen a partir de fotos de las estancias en 360º y no permiten al usuario un desplazamiento libre. Este es el motivo por el que se pretende dar más libertad de movimiento utilizando un modelo 3D.

En este caso la visita se realizará en el templo egipcio más grande fuera de Egipto, el Templo Egipcio de Debod.

Además, este tipo de visitas tiene un contenido fijo que los encargados de realizar la visita han introducido en un momento dado. En situaciones en las que hay una investigación en curso, sería muy deseable poder incorporar nueva información de forma dinámica. El caso que vamos a tratar en este proyecto, que es el Templo de Debod, tiene actualmente un proyecto de investigación liderado por el profesor Miguel Ángel Molinero Polo, con el que se ha colaborado en este trabajo. En la actualidad, por tanto, se siguen haciendo importantes descubrimientos en el templo y, por tanto, se desea poder incorporar dichas aportaciones en tiempo real a la aplicación.

Otra motivación importante en este proyecto está relacionado con la docencia impartida en el Departamento de Prehistoria, Antropología e Historia Antigua de la Universidad de la Laguna. En dicho departamento se estudia directamente este templo en las asignaturas de Historia Antigua o Egiptología, realizándose incluso visitas físicas al templo en compañía de los estudiantes. El presente proyecto se ha realizado también en colaboración con profesores de dicho departamento con la intención de utilizar la herramienta de la visita virtual como apoyo a su didáctica.

1.2. Objetivos

Los objetivos principales de este proyecto corresponden con el itinerario que se va a seguir:

Realismo en la visita: Para recrear el templo con mayor exactitud, se realizó una visita con el propósito de obtener las texturas necesarias partiendo de las fotos tomadas.

Modelado en 3D: Utilizando los planos aportados por el Departamento de Historia Antigua, Medieval y Paleografía y Diplomática de la UAM, se ofrece una reconstrucción a escala real del Templo Egipcio de Debod.

Recreación de visita: Para ofrecer una recreación de calidad, es necesaria la movilidad por el edificio para corregir todos los fallos que puedan haber en el modelo 3D del templo. Por ello es preciso que la velocidad del movimiento y la rotación sean adecuadas para el usuario, al igual que los controles para manejarlos.

1.3. Ciclo de vida del software

Este apartado expone las distintas etapas por las que debe pasar el proyecto para garantizar un producto adecuado. Las fases del ciclo de vida son las siguientes:

Análisis de requisitos: Consiste en documentar las necesidades del usuario en relación con la funcionalidad del software. Es decir, se manifiesta *qué* se debe hacer para llevar a cabo el proyecto. Para ello se siguen una serie de pasos, y estos son:

- Obtener del cliente los requisitos en entrevistas.
- Estudiar dichos requisitos para que sean concretos y no den lugar a interpretaciones ambiguas, además de ser clasificados entre requisitos funcionales y no funcionales.
- Repasar con el cliente que los requisitos establecidos cumplan las necesidades reales del proyecto.

Diseño: Esta etapa se encarga de exponer *cómo* se llevará a cabo los requisitos de la fase anterior organizando en módulos la aplicación.

Implementación: Parte desde la codificación, aunque en este caso partirá desde el modelado en 3D, hasta la certificación de las pruebas del software. Las pruebas que se hagan en esta fase disminuirán el riesgo de futuros problemas en el software que puedan aumentar el coste del producto.

Pruebas: Esta fase se hace cargo de verificar el funcionamiento adecuado del proyecto desarrollado y el cumplimiento de los requisitos dados por el cliente.

Mantenimiento: Expone los tipos que hay: correctivo, adaptativo, perfectivo y preventivo.

El modelo de ciclo de vida seleccionado para este proyecto es *el modelo en cascada*, propuesto en 1970 por W. Royce.

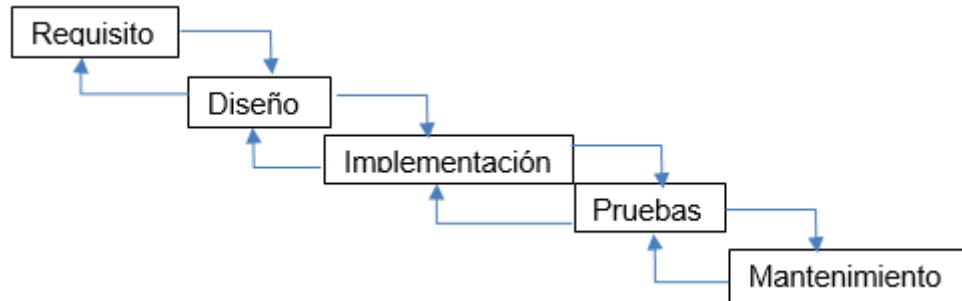


Ilustración 1. Modelo de ciclo de vida en cascada

Se ha escogido el modelo en cascada porque tiene un enfoque rigurosamente ordenado para cada una de las etapas del proceso para el desarrollo del software. Por lo que para pasar a la siguiente fase es necesario haber terminado la anterior, es decir, tener un producto final de cada etapa.

1.4. Estructura del documento.

El proyecto ha pasado por una serie de etapas descritas a lo largo de este documento, relacionado así con los apartados y se verán más adelante.

La sección 2 de este documento, está dedicada al estado del arte. En donde se describe el estudio realizado del tipo de visitas virtuales que ofrecen algunos monumentos, las herramientas de modelado 3D, las herramientas de desarrollo y aplicaciones para el desarrollo del proyecto.

La sección 3 descrita en este documento está compuesta por el análisis de requisitos recogidos de la información dada para el desarrollo del proyecto, por lo que es importante que esté bien definida para evitar problemas en las siguientes fases.

La sección 4 está compuesta por una breve descripción de la obtención del material visual.

El diseño del proyecto se halla en la sección 5, estableciendo la estructura modular tanto para la aplicación de administración de textos como para la aplicación de la visita en 3D, así mismo se expondrán los diagramas diseñados para cada una de ellas.

La fase de implementación de la aplicación y el desarrollo del modelado en 3D se encuentra en la sección 6.

Las pruebas realizadas a la aplicación se encuentran en la sección 7 justificando el comportamiento que debería tener.

La sección 8 y última, estará compuesta por las conclusiones del proyecto y el trabajo futuro.

2. ESTADO DEL ARTE

Este proyecto recreará una visita en 3D por el Templo Egipcio de Debod. Para ello se han realizado distintos estudios, entre ellos está el estudio a las visitas virtuales, es estudio sobre las herramientas de modelado 3D y el estudio de herramientas de desarrollo y de aplicaciones.

2.1. Estudio de visitas virtuales

Estas visitas tienen en común que se han recreado a partir de un conjunto de fotos de 360º de cada estancia.

Así pues, se ha llevado a cabo un estudio de las visitas expuestas en la página oficial del Templo Egipcio de Debod, al igual que las visitas ofrecidas por las páginas del *museo del Romanticismo* y del *museo de Antropología*. Este estudio se ha hecho teniendo en cuenta los siguientes puntos:

- Acceso a las estancias: Se evalúa la distribución de las habitaciones en un menú que aporte al usuario una mayor accesibilidad.
- Información: Se evalúa la calidad de la información expuesta de cada estancia dentro de la visita. También se tendrá en cuenta la cantidad de información que se encuentre.

2.1.1. Visita virtual del Templo Egipcio de Debod

Aspecto de la visita virtual del Templo Egipcio de Debod.



2.1.2. Visita 3D del Templo Egipcio de Debod

El estudio de la visita 3D es el siguiente:



Ilustración 3. Visita 3D Templo Egipcio de Debod.

2.1.3. Visita virtual del Museo del Romanticismo

El análisis de la vista del Museo del Romanticismo es el siguiente:

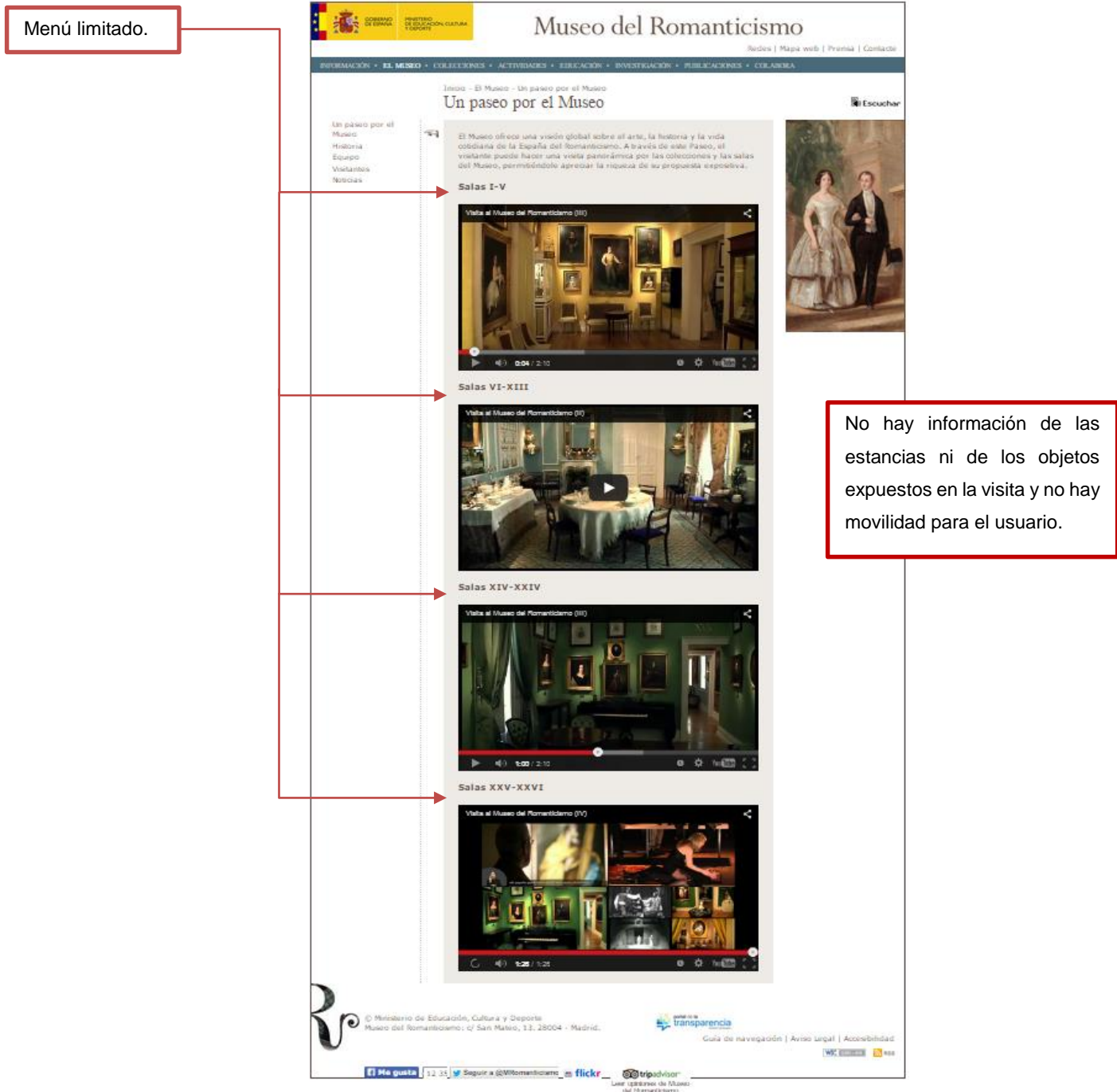


Ilustración 4. Visita museo del Romanticismo

Aunque la información se encuentra disponible en otras secciones de la página, no hay un enlace que las relacione con los objetos que se puedan encontrar en las estancias durante la visita guiada.

2.1.4. Visita virtual del Museo de Antropología

El análisis de la vista del Museo del Romanticismo es el siguiente:



Ilustración 5. Visita museo del Antropología

Da un menú de navegación por las estancias permitiendo mayor movilidad.

El nombre de las salas se hace visible al pasar el ratón por encima de los puntos naranja.

La información de interés para los visitantes es extensa y se encuentra disponible presionando los iconos naranja.

2.1.5. Conclusiones del estudio

Las conclusiones a las que podemos llegar después del estudio son las siguientes:

- La rotación automática de la visita virtual del Templo Egipcio de Debod es demasiado rápida comparada con el movimiento que tiene la visita del museo del Romanticismo.
- La información que ofrecen en sus visitas tanto el Templo Egipcio de Debod como el museo del Romanticismo resulta escasa comparada con la ofrecida por el museo de Antropología.
- El menú que ofrece el Templo Egipcio de Debod combinado con la localización de la estancia no termina de ser tan clara para el usuario como la ofrecida por el museo de Antropología.
- La visita que permite al usuario moverse con mayor libertad es la del museo de Antropología.
- La visita en 3D del Templo Egipcio de Debod no cede al usuario movimiento, ni información textual que complementa a la información visual.

Como se puede apreciar, la combinación de la visita virtual del Templo Egipcio de Debod con la visita del museo de Antropología en una visita en 3D facilitaría al visitante la movilidad por las estancias mientras accede a una información didáctica y extensa de las mismas.

2.2. Herramientas de modelado 3D

Este apartado exhibe los motivos que han llevado a utilizar las herramientas de modelado 3D con las que ha sido desarrollado el proyecto, explicando un poco de cada una de ellas.

2.2.1. Herramientas de diseño gráfico

Para recrear el Templo Egipcio de Debod en 3D, se ha realizado un pequeño estudio sobre herramientas de modelado en 3D disponibles en el mercado. Estas herramientas son Autodesk Maya, Autodesk 3D Max, SketchUp y Blender.

Los puntos que se han tenido en cuenta para evaluarlas son:

- Diseño arquitectónico: Es necesario que con la herramienta se pueda construir un edificio con medidas exactas partiendo de la planimetría.
- Software libre: Se valorará especialmente que la herramienta sea gratuita.
- Texturizado: Una característica imprescindible es que una de las opciones de texturizado sea UV Mapping.
- Interfaz: La interfaz debe ser sencilla e intuitiva.

2.2.1.1. Autodesk Maya

Es uno de los principales programas informáticos dedicados al desarrollo gráfico 3D, usado normalmente por estudios profesionales para modelar, renderizar y especializado en animar. Aunque se pueden crear edificios en este programa, resulta mucho más complejo que en otras herramientas. Tiene una interfaz bastante compleja y poco intuitiva. Una de las características que posee es la opción de texturizar por UV Mapping. Aun así se descartó esta herramienta por ser de pago y no disponer del tiempo necesario para aprender a utilizarlo.

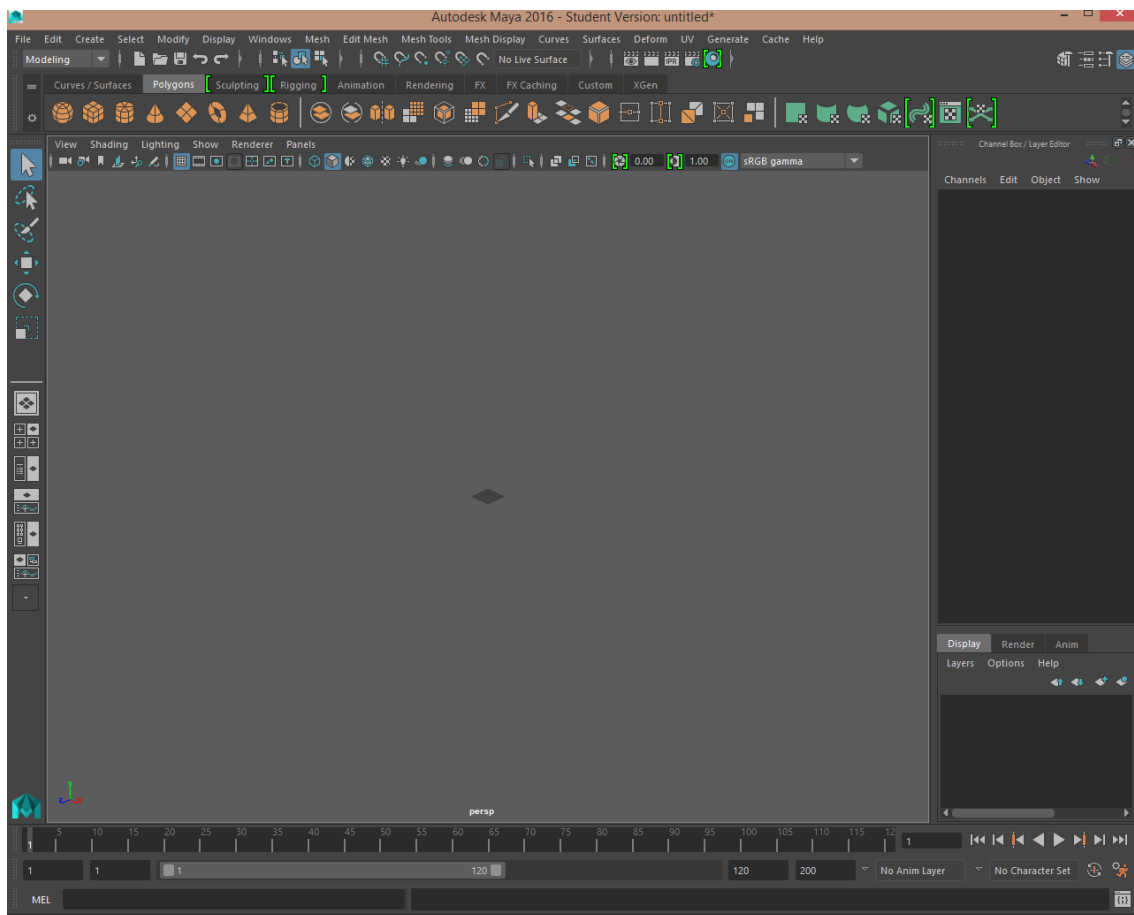


Ilustración 6. Interfaz Autodesk Maya

2.2.1.2. Autodesk 3DS MAX

Es otro gran programa informático dedicado al desarrollo gráfico 3D, usado especialmente para modelar, renderizar y animar. Entre sus características a favor se encuentran la facilidad para el usuario de recrear diseños arquitectónicos y el texturizado por UV Mapping, como contras tiene una interfaz compleja. Puesto que también es un programa de pago y no se disponía del tiempo suficiente para aprender a manejarlo, se descartó esta opción.

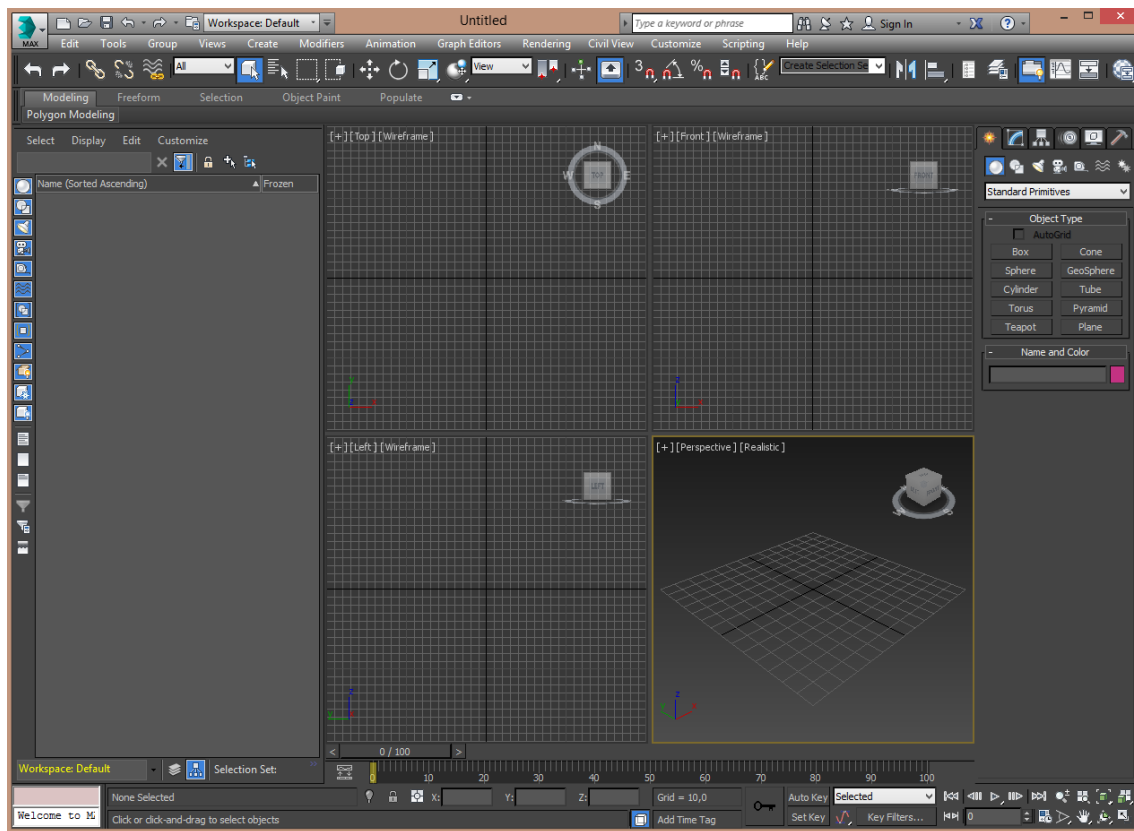


Ilustración 7. Interfaz Autodesk 3D Max.

2.2.1.3. *SketchUp*

Es un programa dedicado especialmente a la creación de edificios. Contiene una interfaz sencilla y posee tanto una versión gratuita como una de pago.

Las formas que ofrece esta herramienta para la creación de objetos son el círculo, el rectángulo, el polígono, la línea, el arco y el trazo a mano alzada. Con estos elementos básicos se pueden hacer diseños de piezas de ingeniería o arquitectura.

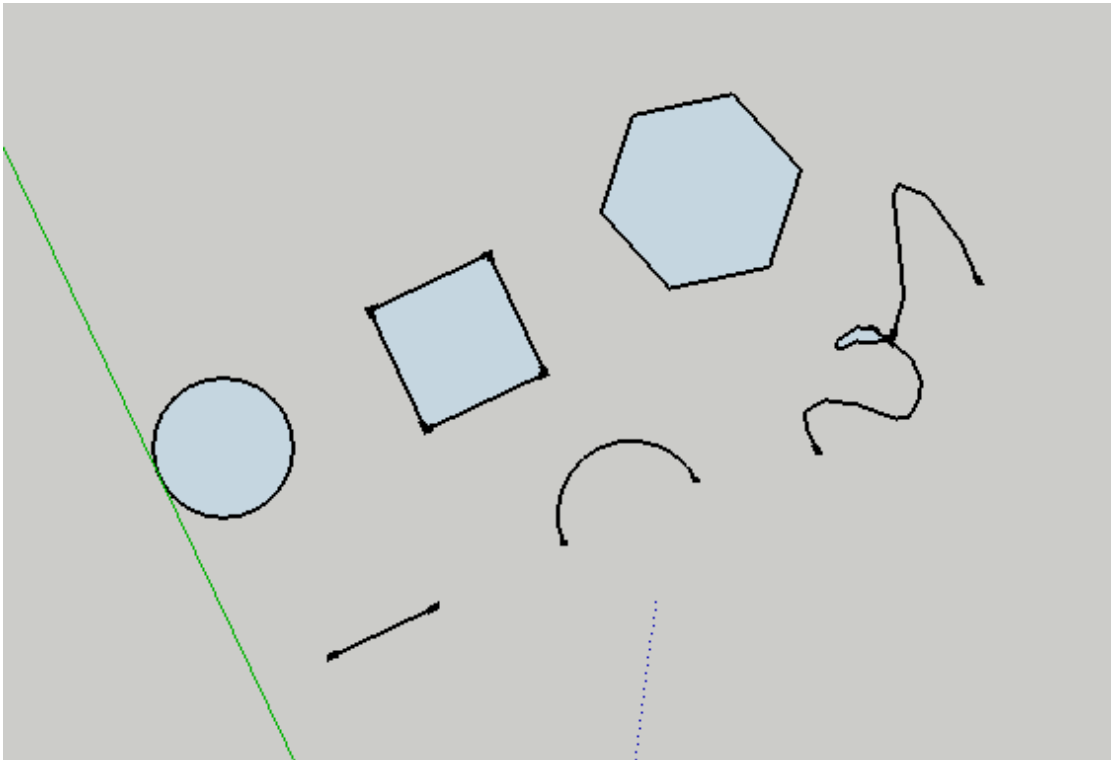


Ilustración 8. Objetos básicos SketchUp

Un inconveniente que se encontró en esta herramienta fue lo poco intuitiva que era al intentar crear la planta del Templo Egipcio de Debod partiendo del plano cedido por el Departamento de Historia Antigua, Medieval y Paleografía y Diplomática de la UAM, con respecto a texturizado UV Mapping es necesario agregar los pluggins correspondientes. Debido a ese motivo y a la interfaz poco intuitiva se descartó esta opción.

Reconstrucción virtual de edificios históricos

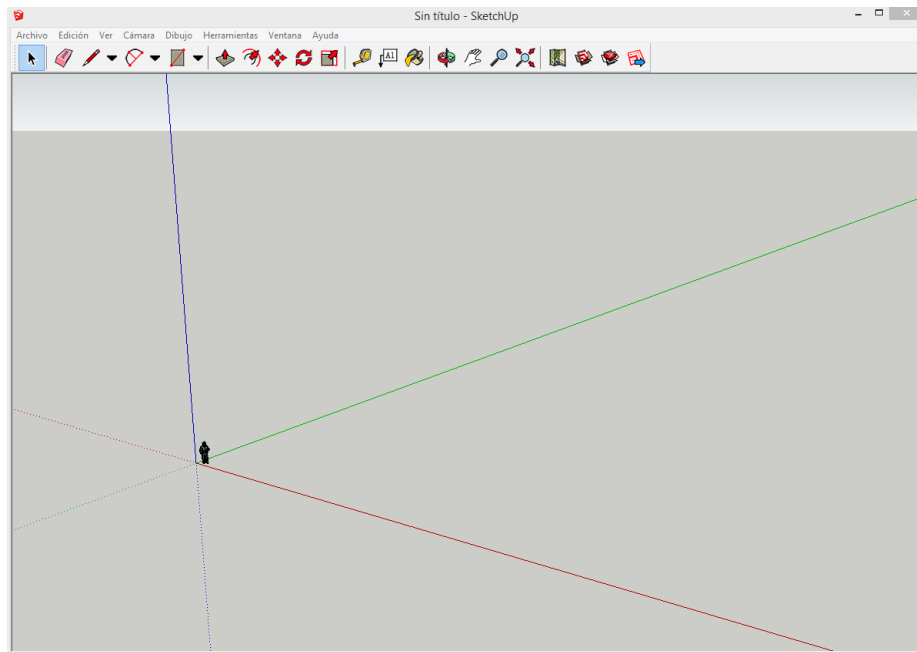


Ilustración 9. Interfaz SketchUp

2.2.1.4. *Blender*

Es una herramienta gratuita especializada en el modelado de objetos en 3D. A estos objetos se les puede añadir texturas, materiales, etc. Entre las herramientas de texturizado se encuentra el UV Mapping que mapea las imágenes sobre los modelos. Además permite recrear edificios con exactitud

Los objetos básicos que se encuentran en Blender son un cubo, un cilindro, un cono, una cabeza de mono, un plano, un toroide, una esfera, una esfera geodésica, un plano subdividido y un círculo.

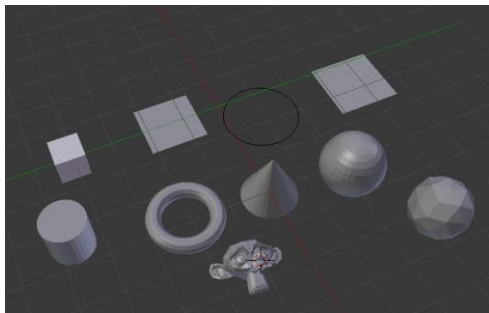


Ilustración 11. Objetos Blender



Ilustración 10. Interfaz Blender

Su interfaz es más compleja que la de SketchUp pero resulta más intuitiva.

2.2.1.5. *Conclusiones de herramientas de modelado 3D*

A partir de este estudio se eligió utilizar Blender como la herramienta de modelado 3D para la recreación del Templo Egipcio de Debod, ya que cumplía con las características que se buscaban, como la recreación de un edificio con medidas reales, el texturizado por UV Mapping, una interfaz intuitiva y ser un programa gratuito.

Además de que tiene una gran cantidad de usuarios activos que facilitan encontrar foros especializados para solucionar las dudas que puedan surgir.

2.2.2. Herramientas de desarrollo y de aplicaciones

Para recorrer el edificio histórico se necesita una herramienta que integre el modelo 3D del Templo Egipcio de Debod, las características que se buscan son las siguientes:

- Ejecutable: La herramienta de desarrollo debe ser capaz de crear un ejecutable de la aplicación en múltiples plataformas.
- Software libre: se valorará que la herramienta sea o tenga una versión gratuita aunque tenga un acceso limitado.
- Desarrollo: la herramienta debe permitir implementación de código y acceso a servidores.
- Acceso: El ingreso a la aplicación debe ser libre sin obligar al usuario a crearse una cuenta.

Los programas que se han estudiado son: Second Life y Unity.

2.2.2.1. *Second Life*



Ilustración 12. Second Life.

Es un universo virtual al que se puede acceder de forma gratuita desde internet. Para poder entrar es necesario registrarse, por lo que se debe ser mayor de edad.

Entre las características que se buscan en la herramienta de desarrollo a la que se desea exportar el modelo 3D del Templo Egipcio de Debod, Second life no cumple con la generación de ejecutables, aunque sea un mundo virtual donde el acceso es gratuito es necesario estar registrado y esto provoca una accesibilidad limitada, obligando a los usuarios a inscribirse en una aplicación que tal vez no sea de su interés. Aunque permite importar objetos, combinarlos con el lenguaje de programación Linden Scripting Language y compartirlos con otros usuarios, Second Life mantiene la mayoría de derechos de autor sobre ellos, y es otra característica por la que se descartó esta opción.

2.2.2.2. Unity

Es una plataforma de desarrollo para crear videojuegos y aplicaciones interactivas multiplataforma tanto en 2D como 3D, por ello genera ejecutables para distintas plataformas. La versión 5 integra la versión gratuita con la de pago, de esta forma se puede acceder a todas las características de Unity que antes no estaban disponibles para todos los usuarios. Entre las condiciones de uso se encuentra que si se desea comercializar el producto creado en esta herramienta de desarrollo ha de pagarse una cantidad acordada.

Permite la comunicación con servidores desde cualquiera de los dos lenguajes de programación en los que permite codificar, C# y Javascript.

Entre las características que se buscaban para la herramienta de desarrollo en la que se desea importar el Templo Egipcio de Debod está el acceso libre a la aplicación, que en este caso Unity cumple.

Entre las desventajas que tiene esta la complejidad de su interfaz, y entre las ventajas está la libertad al programador de realizar cualquier tipo de acción en el mundo que se desea crear.

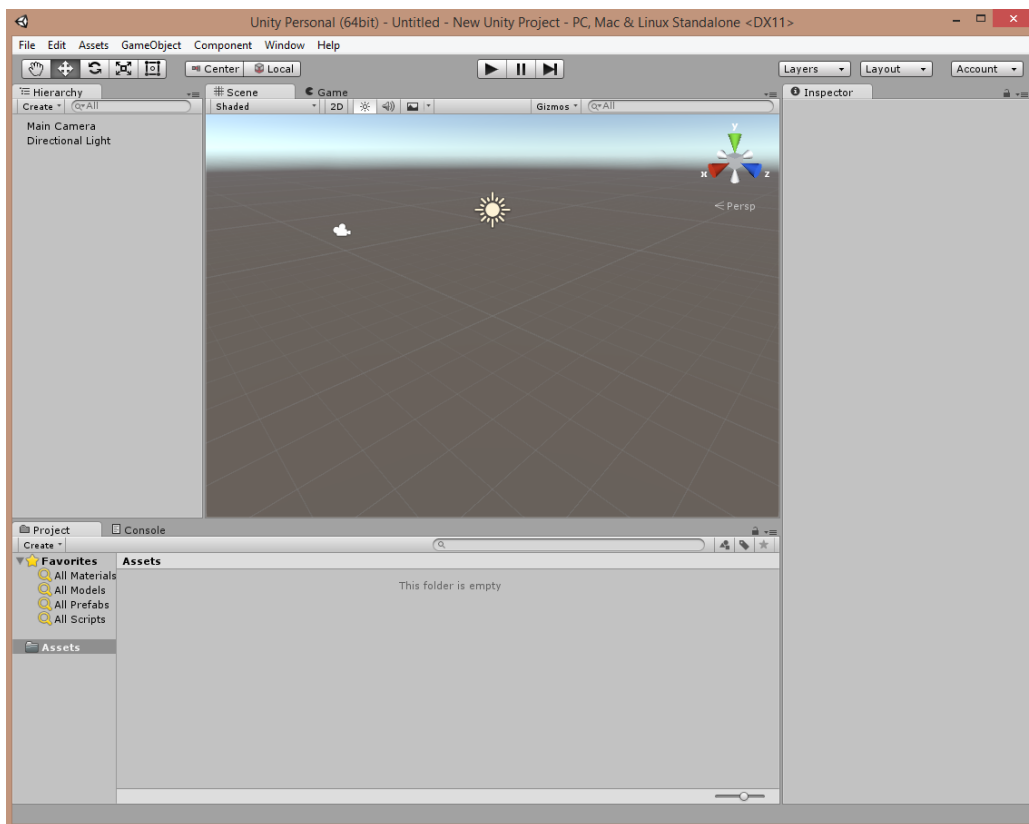


Ilustración 13. Interfaz Unity.

3. ANÁLISIS DE REQUISITOS

El análisis de requisitos está compuesto por una explicación completa de la conducta del software que se va a desarrollar.

Esta sección agrupa todos los requerimientos detallados que debe desempeñar la aplicación para poder recrearlos en las fases posteriores del proyecto.

3.1. Requisitos funcionales

Los requisitos funcionales están agrupados por los tipos de usuario quedarán uso al software.

3.1.1. Administrador

R01	Edición de estancias	El sistema permitirá editar el nombre de las estancias, los ficheros que sirven de información para el visitante y el idioma.
R02	Lista de idiomas	El sistema permitirá al administrador añadir y eliminar los idiomas en los que se encontrarán los textos de la aplicación.

Tabla 1. Requisitos funcionales del administrador.

3.1.1. Visitante

R03	Visita guiada	La aplicación tendrá la cámara en primera persona y seguirá un recorrido por el edificio histórico, deteniéndose en cada estancia donde enseñará el texto correspondiente al usuario y esperará a que éste de la señal de que ha terminado la lectura para continuar con el recorrido.
R04	Visita libre	La aplicación tendrá la cámara en primera persona mientras sigue el recorrido que le indique el usuario. También mostrará una lista de las estancias del Templo Egipcio de Debod.

R05	Menú	El visitante podrá elegir entre la visita guiada, la visita libre y opciones de la aplicación.
R06	Opciones	Las opciones de las que dispondrá el visitante serán: Información de las teclas para desplazarse por el templo. Cambiar velocidad de cámara. Cambiar idioma. Cambiar velocidad de rotación de la cámara. Créditos de la aplicación.
R07	Texto	El visitante tiene textos informativos disponibles en las estancias del templo.

Tabla 2. Requisitos funcionales del visitante.

3.2. Requisitos no funcionales

Los requisitos no funcionales analizados para la aplicación son los siguientes.

3.2.1. Seguridad

R08	Condiciones de la contraseña	La contraseña del administrador contendrá caracteres alfanuméricos, al menos una letra mayúscula y deberá ser mayor o igual 16 caracteres.
R09	Asociación de archivos	Todos los textos subidos estarán asociados al administrador, siendo así de su propiedad.

Tabla 3. Requisitos no funcionales de seguridad.

3.2.2. Interfaz

R10	Interfaz de la aplicación	Recreación del Templo Egipcio de Debod en 3D en un parque rodeado de árboles.
R11	Interfaz para el administrador	El sistema proporcionará al usuario una interfaz sencilla.

Tabla 4. Requisitos no funcionales de interfaz.

3.2.3. Portabilidad

R12	Disponibilidad	La aplicación debe ser soportada en múltiples plataformas: PC, Mac, Linux.
------------	----------------	--

Tabla 5. Requisitos no funcionales de portabilidad.

3.2.4. Rendimiento

R13	Características	<p>La aplicación debe poder ser ejecutada de forma óptima en un ordenador de las siguientes características:</p> <p>Disco duro: mínimo un 1GB libre de memoria.</p> <p>RAM: 4GB</p> <p>Procesador: AMD E-450 APU with Radean(tm) HD Graphics 1.65 GHz o similar</p> <p>Tarjeta gráfica: Dual core and amd radeon graphics o similar.</p>
------------	-----------------	--

Tabla 6. Requisitos no funcionales de rendimiento.

3.2.5. Usabilidad

R14	Navegación por estancias	La aplicación tendrá un menú con las estancias del edificio histórico.
R15	Botón de volver	La aplicación tendrá la opción de volver al menú desde cualquier parte presionando el botón Esc.
R16	Identificación	El sistema permitirá al usuario ingresar a la aplicación de gestión de archivos con un usuario y una contraseña.
R17	Gestión	El sistema debe permitir al administrador gestionar los archivos

Tabla 7. Requisitos no funcionales de usabilidad.

3.2.6. Accesibilidad

R18	Movimiento de traslación	<p>Las teclas que se encargarán de desplazar al visitante por el edificio son WASDQE.</p> <p>W: Avanza hacia adelante.</p> <p>A: Desplazamiento hacia la izquierda.</p> <p>S: Desplazamiento hacia atrás.</p> <p>D: Desplazamiento hacia la derecha.</p> <p>Q: Desplazamiento hacia abajo.</p> <p>E: Desplazamiento hacia arriba.</p>
R19	Movimiento de rotación	<p>Las teclas que se encargarán de girar la visión del usuario son ArrowUp, ArrowLeft, ArrowRigth, ArrowDown.</p> <p>ArrowUp: rotación sobre el objeto hacia arriba.</p> <p>ArrowLeft: rotación sobre el mundo hacia la izquierda.</p> <p>ArrowRigth: rotación sobre el mundo hacia la derecha.</p> <p>ArrowDown: rotación sobre el objeto hacia abajo.</p>

Tabla 8. Requisitos no funcionales de accesibilidad.

4. DISEÑO DEL SOFTWARE

Este capítulo define el proceso de las especificaciones formales que dejarán a los desarrolladores las referencias que necesitan para la etapa de desarrollo del producto. De manera que esta sección está compuesta por la arquitectura del software correspondiente a las dos aplicaciones que componen el proyecto.

4.1. Diseño general

Este proyecto se compone de dos aplicaciones por lo que necesitará dos diseños, uno para el administrador de la aplicación y otro para la aplicación principal.

El diseño general compuesto por ambas aplicaciones es el siguiente:

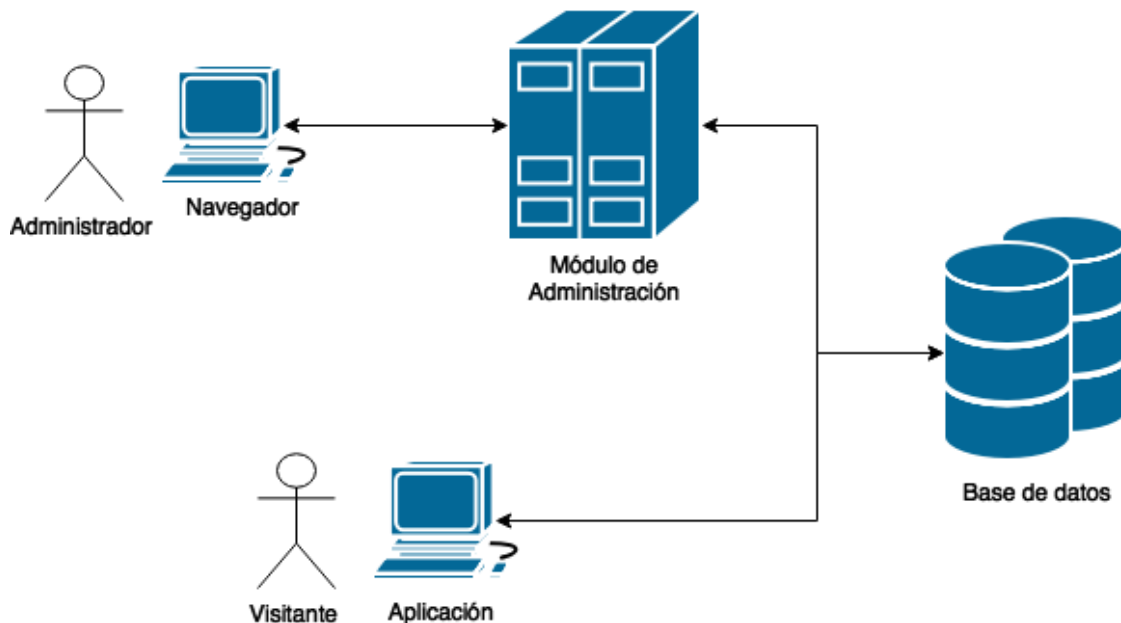


Ilustración 14. Esquema general del sistema

Donde la aplicación para la administración de ficheros de textos se conecta con el módulo de administración que a su vez se conectará con la base de datos para asegurarse de que la identificación es correcta. Desde el lado del administrador se cargarán los ficheros de texto en la base de datos pasando por el módulo de administración.

La aplicación de la visita, a la que tendrá acceso el visitante, hará peticiones a la base de datos para enseñar los textos de las estancias al usuario.

4.2. Diseño modular de la visita 3D

La aplicación destinada a los visitantes del templo virtual se ha dividido en los siguientes módulos:

- Gestor de escenas.
- Gestor de movimiento.
- Gestor del menú de opciones.
- Gestor del transporte.
- Gestor del tour.
- Gestor de ventanas emergentes.
- Gestor de acceso a base de datos.

Esta división permite tener una mayor modularidad, que aumentará considerablemente la mantenibilidad y extensibilidad del sistema. Además permite intercambiar módulos fácilmente en función de las preferencias del usuario.

El siguiente diagrama muestra las distintas dependencias entre los módulos:

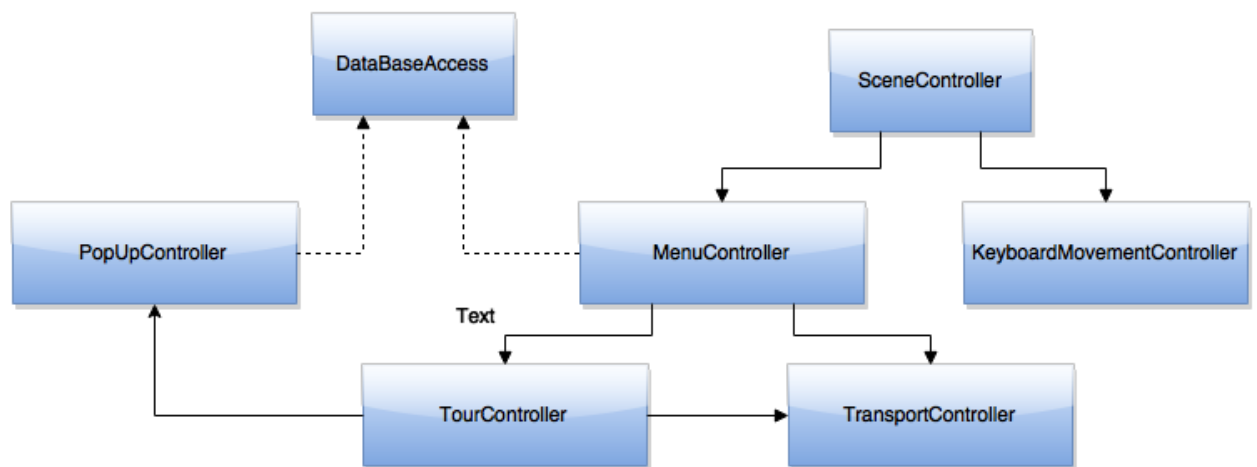


Ilustración 15. Esquema general de la aplicación.

A continuación se explicará con mayor detalle las funciones de cada módulo y sus interdependencias.

4.2.1. Gestor de escenas

Este módulo es el principal de la aplicación. Sirve como referencia del resto de módulos y como receptor de los inputs introducidos por el usuario. Se encarga así mismo de inicializar el resto de módulos al comenzar el programa.

Hace uso del módulo MenuController para gestionar tanto el menú inicial como el menú de opciones en cualquier momento de la ejecución de la simulación. Debe encargarse de bloquear el movimiento autónomo del jugador, bloqueando el módulo KeyboardMovementController, cuando el menú esté disponible.

También gestiona el módulo de movimiento, KeyboardMovementController, activándolo y desactivándolo según corresponda.

4.2.2. Gestor de movimiento

Este módulo se encarga de mover al personaje principal a través del escenario. Se activa y desactiva desde el controlador principal, SceneController, en función de si el usuario debe estar bloqueado de movimiento (menús, visitas guiadas...) o no.

Inicialmente se contará con un único controlador de movimiento por teclado, pero se planea ampliar teniendo distintos controladores para ser usados en función de las preferencias del usuario.

4.2.3. Gestor del menú de opciones

Este módulo se encarga de mostrar y gestionar el menú, ya sea el inicial o el menú de opciones de la aplicación. El SceneController es el encargado de activar o desactivar este módulo.

Hace uso del módulo de acceso a la base de datos, DataBaseAccess, para obtener los idiomas disponibles a elegir y los textos a mostrar en cada idioma.

Dado que estos menús se encargan de activar el tour o el transporte a través de la aplicación, este módulo necesita de los dos módulos que se encargan de ello: TourController y TransportController.

4.2.4. Gestor del transporte

Este módulo se encarga de transportar al personaje de un punto a otro del templo. Se ha independizado del módulo de Tour porque se han considerado varias formas de transporte: instantáneo, desplazamiento rápido, etc. Esta modularidad ofrece la posibilidad de cambiar el tipo de transporte básico sin necesidad de cambiar el módulo de Tour.

Este módulo se usa directamente cuando se selecciona la opción de ser transportado en el menú. También se utiliza indirectamente al seleccionar el Tour.

4.2.5. Gestor del tour

Este módulo gestiona la visita guiada completa a través del templo. Esta opción está disponible desde el menú, por lo que será el módulo MenuController quien lo active cuando se seleccione.

Hará uso del TransportController para desplazar al visitante o usuario a las distintas localizaciones del templo, y del PopUpController para mostrar los textos relevantes de cada una de estas situaciones.

Este módulo informa al módulo SceneController al comenzar el Tour para que bloquee el módulo de movimiento, impidiendo el desplazamiento autónomo del usuario durante la duración de la visita.

4.2.6. Gestor de ventanas emergentes

Este módulo gestiona los mensajes informativos que aparecerán en la visita al templo mostrando información adicional al visitante.

Hace uso del módulo de acceso a base de datos, DataBaseAccess, para obtener los textos en el idioma que el usuario haya seleccionado en las opciones de la aplicación.

Este módulo se encarga también de dar la opción al usuario de ver un texto cuando se encuentra en una zona con información extra disponible.

El módulo de Tour, TourController, hace uso de este módulo de forma automática para mostrar los mensajes durante el Tour.

4.2.7. Gestor de acceso a base de datos

Este módulo proporciona al resto de módulos configuraciones guardadas en base de datos. Estas configuraciones incluyen los textos en distintos idiomas para las opciones del menú de MenuController y los mensajes informativos de PopUpController.

Tener este acceso a la base de datos en un módulo independiente permite poder cambiar el modo de almacenamiento de estas configuraciones en cualquier momento con cambios mínimos, sino nulos, en el resto de módulos.

4.3. Diseño aplicación de administración

La aplicación destinada al administrador se ha dividido en los siguientes módulos:

- ViewLogin.
- ViewMenu.
- ViewInstance.
- RequestController.
- ServerController.

Esta división permite tener una mayor modularidad, que aumentará considerablemente la mantenibilidad y extensibilidad del sistema.

El siguiente diagrama muestra las distintas dependencias entre los módulos:

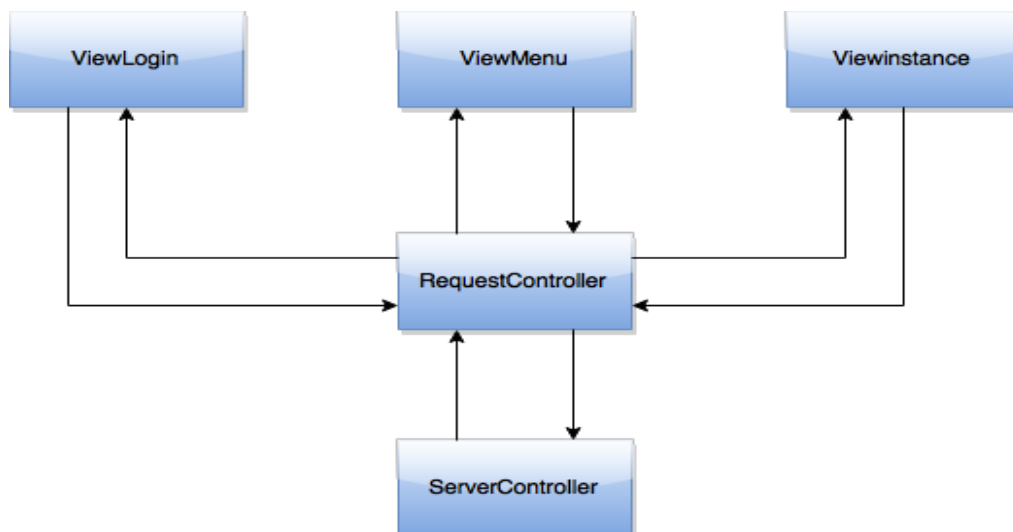


Ilustración 16. Esquema de la aplicación del administrador.

A continuación se explicará con mayor detalle las funciones de cada módulo y sus interdependencias.

4.3.1. ViewLogin

Esta es la vista de la identificación del administrador. Hace uso del módulo encargado de las peticiones a servidor, RequestController, para reconocer al administrador.

4.3.2. ViewMenu

Este módulo es la vista del menú del administrador, contiene la lista de las estancias del templo. Esta lista la obtiene usando el módulo de peticiones, RequestController, para comunicarse con la base de datos.

4.3.3. ViewInstance

Este módulo es la vista de la instancia elegida en el menú. Contiene la edición de ficheros de texto, de idioma, etc. Utiliza el módulo de peticiones, RequestController, para enviar los datos a la base de datos.

4.3.4. RequestController

Este módulo en javascript que gestiona las peticiones de las vistas a la base de datos usando el módulo de servidor, ServerController.

4.3.5. ServerController

Módulo en PHP que hace las consultas a la base de datos, y se comunica con las vistas por medio de RequestController.

4.4. Estructura de la base de datos

La base de datos contará con 2 tablas, Language y Text.

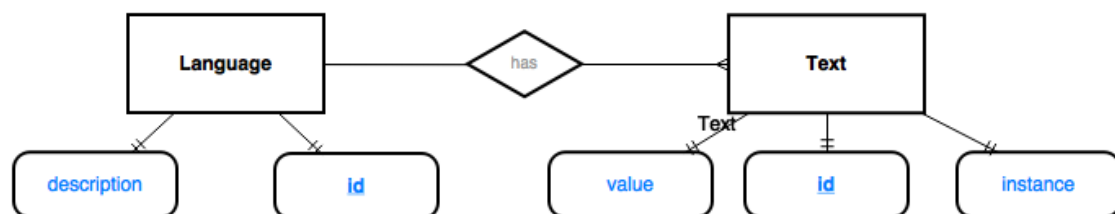


Ilustración 17. Estructura de la base de datos.

4.4.1. Tabla language

La primera de ellas, Language, contará con la lista de idiomas en que está disponible la aplicación en un momento concreto. Esta tabla cuenta con:

- **Id:** Corresponde con el identificador que se necesitará para realizar consultas sobre Text.
- **Description:** Una breve descripción para cada idioma, que se mostrará a los usuarios de la aplicación al ir a seleccionarlo.

4.4.2. Tabla Text

Por otro lado, la tabla Text contendrá los textos que se mostrarán a los usuarios en función del idioma seleccionado. Esta tabla cuenta con los siguientes campos:

- **Id:** Un identificador del texto, preferiblemente descriptivo con respecto a su situación. Por ejemplo, el texto que podrá verse al observar el muro norte de la sala inicial del templo podría denominarse hall.north.1. Este identificador será usado por la aplicación para consultar el texto a mostrar.
- **Language:** Identificador que referencia a la tabla Language. Al seleccionar el usuario un idioma, todas las consultas tras la elección se realizarán con el identificador del lenguaje en cuestión, cambiando por completo el idioma de los textos mostrados en la aplicación al idioma seleccionado.
- **Value:** Valor a mostrar. Inicialmente este valor constará de un texto plano a introducir en el popup correspondiente. Posteriormente se considerará la admisión de formato a través de HTML.
- **Instance:** Nombre de la sala donde se mostrará el texto. El objetivo de este campo es poder facilitar la administración de los textos al administrador, pudiendo realizar un filtro básico al buscar un texto concreto.

5. IMPLEMENTACIÓN

Este capítulo describe el proceso de desarrollo tanto de la aplicación de la visita virtual como la aplicación para el administrador.

5.1. Modelado 3D del templo

Partiendo del estudio realizado de las herramientas de modelado 3D que se han visto en el apartado 2.2, se decidió que el programa adecuado para recrear el templo era Blender.

Utilizando la planimetría cedida por el Departamento de Historia Antigua, Medieval y Paleografía y Diplomática de la UAM, se obtuvieron las medidas del templo.

Para comenzar con la reconstrucción del templo se plasmó en Blender la planta, usando planos. Los planos en Blender son objetos básicos compuestos por cuatro vértices, cuatro aristas y una cara.

Extendiendo las aristas adecuadas sobre el eje Z de los planos se levantaron las estancias desde el alzado posterior al alzado principal. A partir de los planos alzados se ajustaron los ángulos de las paredes, por otro lado, se reconstruyeron los pilonos de explanada frontal y las columnas de la entrada de forma independiente, para ser incorporados al templo, completando la estructura.

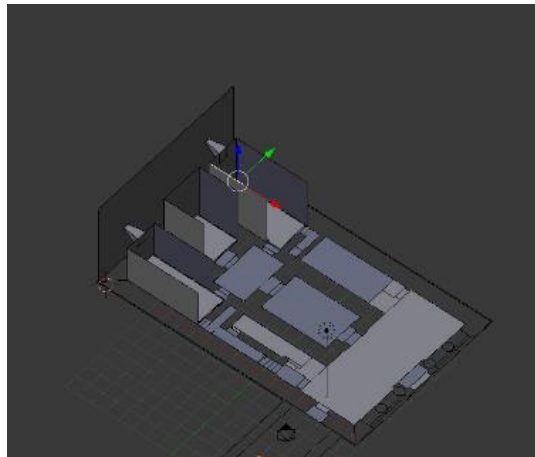


Ilustración 18. Alzamiento del templo en Blender.

Reconstrucción virtual de edificios históricos

Una vez creado el modelo en 3D del templo se recorrió cada estancia desde las cámaras de Blender para corregir los fallos que se hubieran podido producir en la reconstrucción.

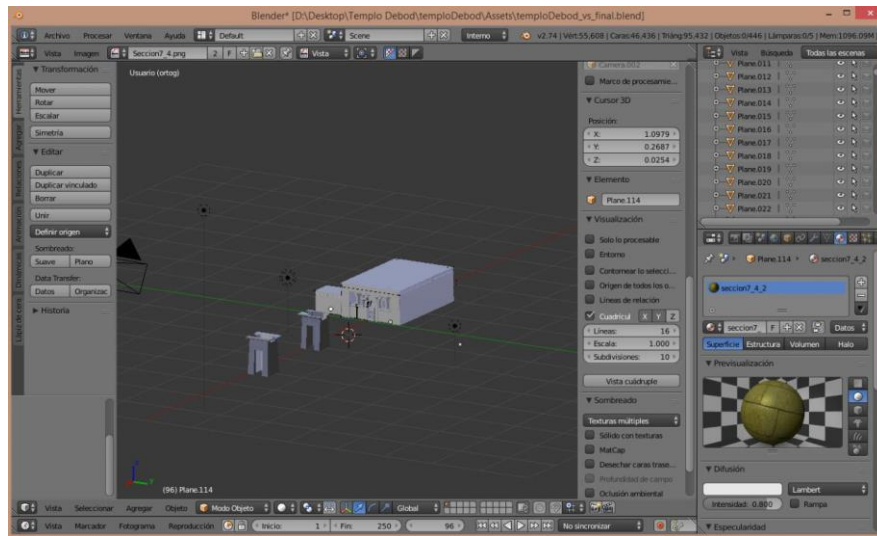


Ilustración 19. Modelo 3D del Templo Egipcio de Debod.

Por último, se completó el templo añadiendo la plataforma exterior sobre la que reposa.

5.2. Texturizado

El objetivo de este proyecto es ofrecer una visita guiada de la máxima calidad y fidelidad a los usuarios de la aplicación, por lo que el correcto texturizado del modelo creado en Blender es de vital importancia.

No solo es necesario obtener las texturas adecuadas, sino que la iluminación y el efecto de relieve de las mismas tengan la calidad que se espera de una visita de estas características.

Para darle al modelo este aspecto realista se usaron una serie de herramientas de tratamiento de imagen del modo que describiremos a continuación.

Como primer paso se crearon las texturas en el programa Manga Studio 5 partiendo de las fotos obtenidas de la visita realizada al Templo Egipcio de Debod.

Manga Studio 5 ofrece una amplia variedad de herramientas de manejo de imágenes, gracias a las cuales se ha podido asignar de manera fácil y eficaz las texturas a cada una de las piedras de una pared de forma individual.

Se importaron todas las imágenes correspondientes a la planimetría, donde se seleccionaron cada una de las piedras del templo que iban a ser texturizadas.

Reconstrucción virtual de edificios históricos

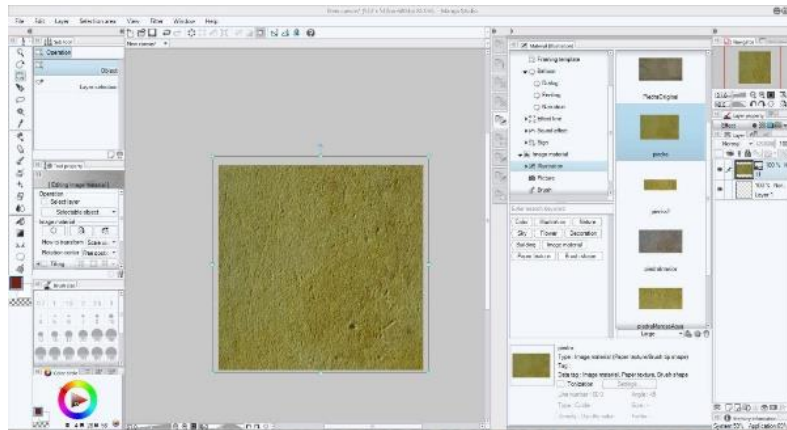


Ilustración 21. Creación de textura en Manga Studio 5.0

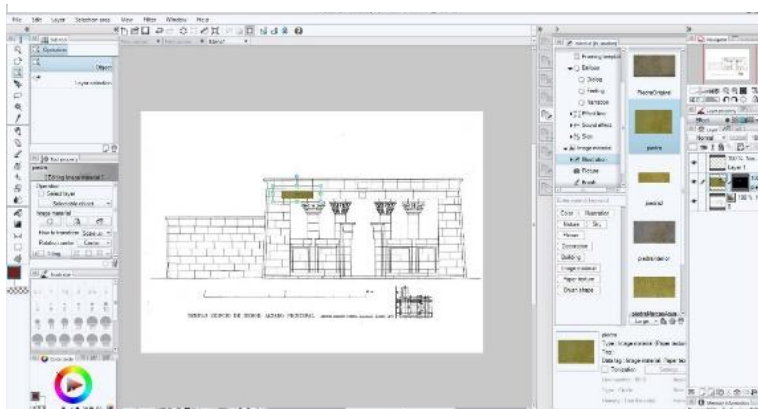


Ilustración 20. Creación de textura del alzado principal del templo en Manga Studio 5.0

Una vez creadas las texturas para cada uno de los muros del templo, se usó Blender para asignar esas texturas al modelo 3D del templo. Para ello se aplicó la técnica UV Mapping. Esta técnica permite asignar a cada cara seleccionada una textura completa o una parte de la textura, de este modo se pudo redimensionar cada cara al tamaño de la textura que le correspondía.

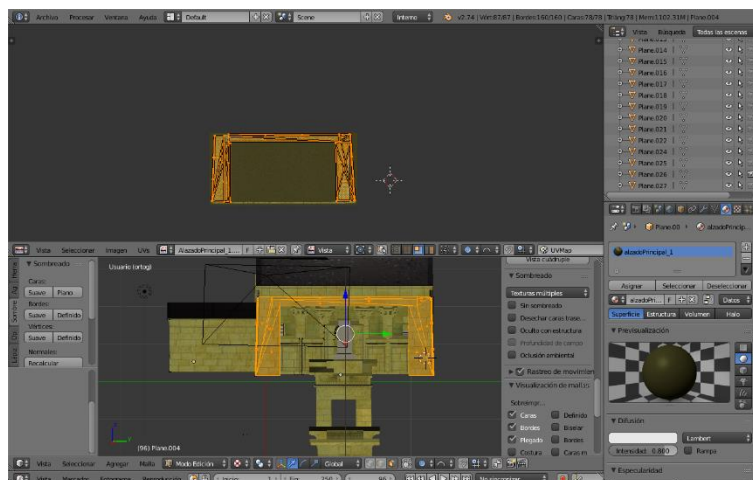


Ilustración 22. Texturizado por UV Mapping en Blender.

Para las texturas de la plataforma del templo, sólo se trataron las fotos sacadas del suelo del exterior del templo que se obtuvieron en la visita.

5.3. Integración del modelado 3D en Unity

Una vez completado el templo en Blender con sus texturas, se procedió a trasladarlo a Unity, donde se completaría la aplicación. Para ello se exportaron los modelos en formato BLEND. Este formato contiene las dimensiones de las caras que corresponden con las texturas.

Puesto que Unity no carga las texturas del modelo hay que volver a asignarlas a cada plano, aun así el formato en el que se encuentran guarda las posiciones en las que se debe situar la cara del plano sobre la textura, por lo que no hay necesidad de realizar más acciones sobre las texturas.

El sistema de materiales de Unity permite usar un relieve para las texturas, proporcionando un efecto final más realista. Para crear estos relieves, se usó otro programa de tratamiento de imágenes, GIMP 2.

GIMP 2 cuenta con una herramienta de mapeo de relieve que crea estos materiales.

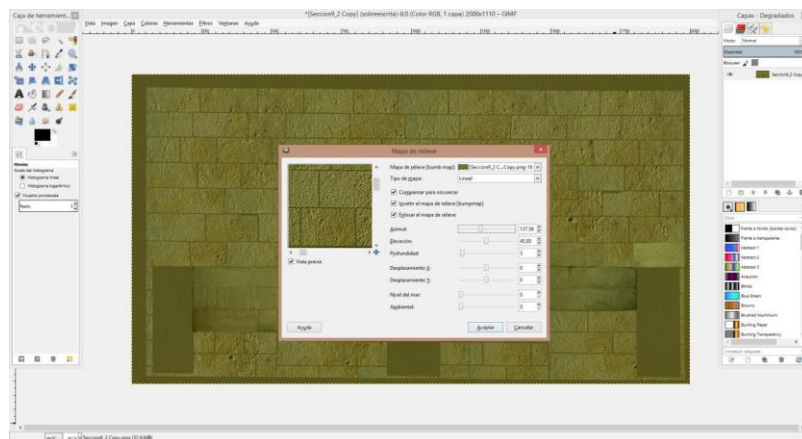


Ilustración 23. Creación de materiales en GIMP 2.

Con el templo y la plataforma importados con éxito en Unity junto con sus texturas, se completó el entorno incorporando diversos efectos.

Entre los efectos que se han agregado se encuentra el agua. El agua en Unity era una característica de pago hasta la versión 4, debido a ello tiene cualidades como reflejar los objetos con los que tiene contacto y moverse según el viento aplicado.

Además se ha añadido una fuente de agua creada a partir de partículas, estas partículas tienen una cantidad delimitada y un tiempo de vida corto para crear el efecto deseado.

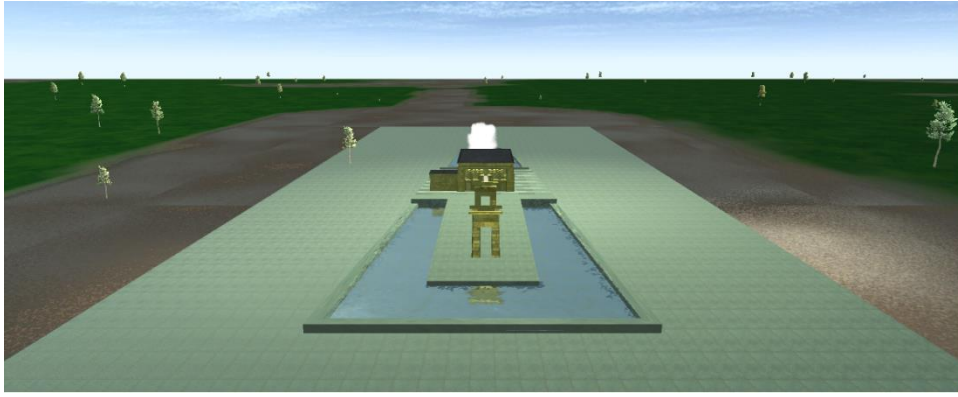


Ilustración 24. Templo Egipcio de Debod integrado en Unity.

5.4. Implementación de la aplicación en Unity

El uso de la herramienta de desarrollo Unity gira en torno a un escenario donde se pueden situar los objetos que componen el entorno interactivo a crear. Estos objetos pueden estar compuestos de las formas básicas de Unity, o pueden ser importados como es el caso del templo. Cada uno de estos objetos cuenta con numerosas propiedades que los definen: tamaño, posición, rotación, peso, fricción, material de renderizado, características de la colisión con otros objetos...

Adjuntando scripts a estos objetos, Unity permite modificar estas características en tiempo de ejecución, creando la aplicación interactiva que se busca.

También permite crear scripts tanto en C# como en JavaScript. Se ha elegido JavaScript porque se tiene un mayor dominio sobre él, ya que se ha estudiado a lo largo del grado y se ha utilizado durante las prácticas de empresa.

Al adjuntar un script a un objeto, desde el script se puede acceder automáticamente a todas las características del mismo, a través del objeto central de Unity, el objeto `GameObject`. Adicionalmente, Unity define un amplio catálogo de funciones que, de declararse en el script, serán invocadas por el motor de Unity cuando suceden determinados eventos. Por ejemplo, la función `Start` se invocará al comienzo de la vida del script, la función `Update` se invocará en cada frame del juego, y la función `OnMouseOver` se ejecutará cuando el cursor pase sobre el objeto.

A continuación explicaremos con detalle cada uno de los módulos de la aplicación, y los scripts que los componen.

5.4.1. Gestor de escenas

El módulo de gestión de escenas es clave en toda aplicación, pues dirige el desplazamiento entre los distintos escenarios de los que se compone. En este caso, solo se cuenta con un único escenario, por lo que este módulo se encarga de la coordinación entre los distintos estados que puede tener la aplicación.

Este módulo tiene un único script, denominado `SceneController`. Este script debe estar presente en todo momento, por lo que se situará en el objeto cámara de la escena para asegurarse de ello.

Puesto que debe asegurarse de que el resto de módulos y elementos se encuentran ejecutándose al comenzar la aplicación, los instanciará al comenzar. En su función `start` adjuntará los scripts necesarios a la cámara a través de la función `GameObject.AddComponent()`.

Una vez iniciados todos los módulos, debe coordinarlos siguiendo las siguientes reglas:

- Inicialmente, se mostrará el menú.
- Siempre que el menú se encuentre visible, el módulo de control del movimiento se encontrará desactivado.
- Durante un tour automático, el módulo de control del movimiento se encontrará desactivado.
- El menú se activará y desactivará con la tecla `Esc`.
- El cursor solo estará disponible en el menú.

Para saber cuándo se ha pulsado la tecla `Esc`, en la función `Update`, que se ejecuta en cada frame, consultará activamente el valor de `Input.GetKey(KeyCode.Escape)`, reaccionando acorde a su valor.

5.4.2. Gestor de movimiento

El módulo de gestión de movimiento se encarga de mover al personaje reaccionando a los inputs del usuario. En este caso, los controles elegidos son los siguientes:

- `W`, `A`, `S`, `D`, `E`, `Q`
- `ArrowUp`, `ArrowRigth`, `ArrowLeft`, `ArrowDown`

Dada la simplicidad de los controles, que hace innecesario, por ejemplo, el procesamiento extra que implicaría tratar con la posición del cursor para girar la cámara, todo el comportamiento de este módulo se concentra en el script `KeyboardMovementController`.

Para implementar esto, se consulta activamente en la función `Update` si las teclas correspondientes han sido pulsadas, al igual que se hacía en el gestor de escenas para saber cuándo activar y desactivar el menú.

Este gestor de movimiento debe estar desactivado a voluntad del gestor de escenas, por lo que la lectura de esos inputs depende de un boolean que controla el `SceneController`.

5.4.3. Gestor del menú de opciones

El gestor del menú de opciones se encarga tanto de mostrar el menú como de reaccionar a cuando se seleccionan las opciones. Para ello, Este módulo se divide en una serie de scripts que facilitan su manejo:

- **LanguageController:** Este script se encarga de mostrar el submenú de idiomas y reaccionar a sus estímulos. Hace uso del conjunto de funciones de gestión de la interfaz que ofrece Unity.
- **CreditController:** Este script, al ser activado, muestra los créditos de la aplicación en el centro de la pantalla.
- **TourController:** Este script muestra los dos botones principales del menú de opciones, el botón de comienzo de Tour completo y el botón de selección de escenas. Cuando alguno de los dos es pulsado, este script se encarga de informar a `MenuController`, script del que hablaremos a más adelante.
- **RoomSelectionController:** Este script muestra y gestiona la selección de salas. Cuando una es seleccionada, se lo comunica a `MenuController`.
- **ConfigurationController:** Este script muestra las opciones de configuración de la aplicación, entre las que se incluyen la sensibilidad del giro de la cámara y su velocidad.
- **MenuController:** Este script constituye el centro de este módulo, y se encarga de coordinar el resto de script activándolos y desactivándolos según corresponda. Adicionalmente se encarga de mostrar y gestionar directamente

los botones extra como el botón de mostrar los créditos, el de mostrar las opciones de configuración y el de salida de la aplicación.

La lógica de estos scripts, que principalmente gestiona MenuController, se resume en el siguiente esquema:

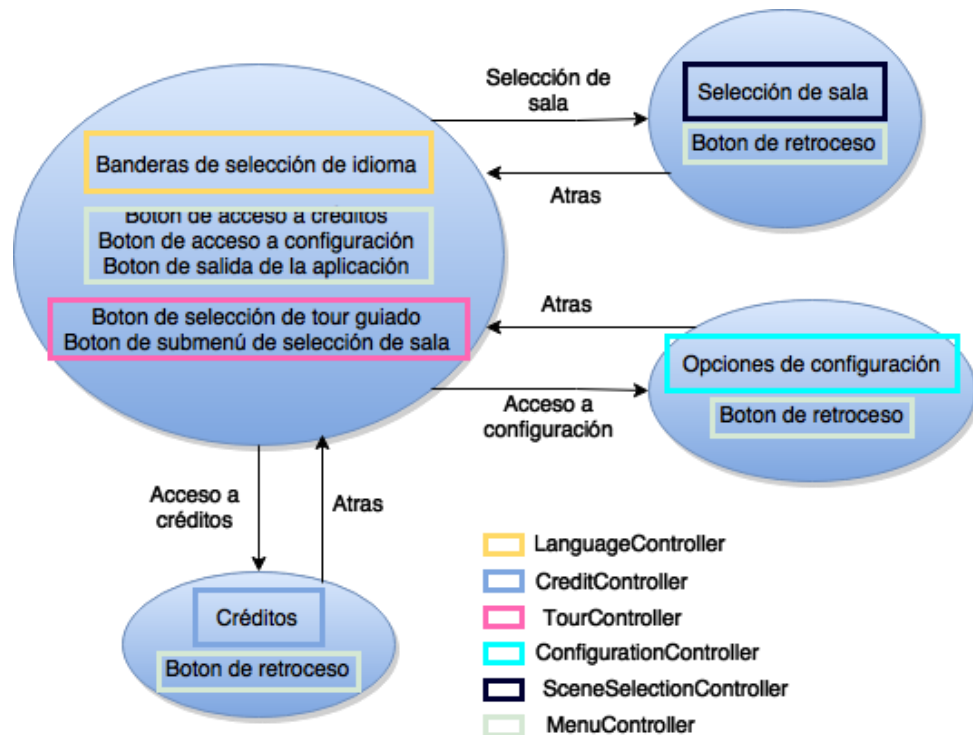


Ilustración 25. Esquema de flujo del menú de la aplicación.

SceneController controla la aparición/desaparición del menú a través de MenuController, que sirve de fachada del módulo al resto del sistema.

5.4.4. Gestor del transporte

El módulo de gestión del transporte se encarga de desplazar al jugador entre las distintas salas del templo cuando éste lo desee. Este módulo se activa desde MenuController, cuando se desea ir a una sala concreta, y desde TourController (que veremos más adelante que pertenece al módulo de gestión del tour), cuando se desea realizar el tour completo.

Aunque durante el diseño del sistema se consideró un módulo independiente por la complejidad que entraña este desplazamiento, al implementarlo se vio innecesario el

uso de más de un script. Este módulo, por lo tanto, incluye solo el script `TransportController`.

El objetivo final de este módulo es que dada una sala destino, el módulo traslada al visitante de forma fluida hasta dicha sala. Para implementar lo se creó un sistema de enrutamiento similar al usado en internet, usando una serie de puntos de referencia distribuidos por todo el templo.

`TransportController` contiene una tabla (array multidimensional) con los siguientes campos: `actualCoordinates`, `objectiveRoom`, `objectiveCoordinates`.

Esta tabla contiene la posición a la que se debe ir para acabar alcanzando la sala objetivo desde una posición concreta. Estas posiciones de origen y destino corresponden con las coordenadas de los distintos puntos de referencia mencionados.

A partir de esta tabla, el script `TransportController` desplaza al personaje siguiendo el siguiente algoritmo:

1. destino \leq `actualCoordinates` más cercanas a la posición inicial
2. mientras no lleguemos a destino
 - a. nos movemos hacia destino
3. destino \leq consultamos en tabla nuevo destino
4. si la tabla no contiene el destino
 - a. hemos llegado
5. en caso contrario
 - a. volvemos al paso 2

La complejidad de esta solución se encuentra en el tamaño de dicha tabla, que se ha decidido cargar desde fichero para facilitar la gestión de la misma.

5.4.5. Gestor del tour

Este módulo se encarga de gestionar el modo tour automático de la aplicación, haciendo uso del módulo de gestión de mensajes y el módulo de gestión de transporte que acabamos de ver.

Una vez más, este módulo se contiene en un único script, `TourController`.

Este script, una vez activado por el módulo de gestión del menú de opciones, bloquea, a través de `SceneController`, el gestor de movimiento del visitante. Tras esto, hace uso

de `TransportController` y de `PopUpController` (script perteneciente al módulo de gestión de ventanas emergentes que veremos más adelante) para llevar a cabo una visita guiada siguiendo unos periodos de tiempo concretos entre acción y acción. La configuración de estos periodos de tiempo se encuentra en otro fichero de configuración del sistema.

Una vez terminado el tour, devuelve el control al visitante.

5.4.6. Gestor de ventanas emergentes

Este módulo se encarga de gestionar la aparición de ventanas con textos en la aplicación. Este gestor está compuesto por el script `PopUpController`, que gestiona los textos de la aplicación, y el script `PopUpTrigger`, que se adjuntará a objetos invisibles (objetos con el componente de renderizado desactivado) marcados como triggers.

Un trigger es un evento que se dispara cuando hay un contacto entre un objeto y el objeto marcado como trigger. En este caso, en vez de haber una colisión se llama a una serie de funciones en el script que está adjunto al objeto, como lo son `OnTriggerEnter`, `OnTriggerExit` y `OnTriggerStaty`.

Se marcarán objetos invisibles como trigger, estos objetos tendrán un estado por defecto que será *desactivado*. Este estado cambiará cuando el usuario colisione con el objeto que llamará a `OnTriggerEnter` de `PopUpTrigger`, este método cambiará el estado del trigger y lo pondrá en *activado*, mostrando el mensaje '*Pulse la tecla X para más información*' en una ventana emergente.

En el momento en que dicha tecla se pulse mientras se permanezca en el trigger, se invocará al método `showText` de `PopUpController`. Esta función recibe un código en su invocación. Con este código y con el idioma activo, obtenido a través de `SceneController`, se utilizarán para realizar una petición al módulo de acceso a base de datos para obtener el texto y mostrarlo.

Este texto desaparecerá cuando se vuelva a pulsar la tecla correspondiente. También desaparecerá, pero de forma automática, en el caso de que el objeto cámara manejado por el usuario deje de estar en contacto con el objeto trigger.

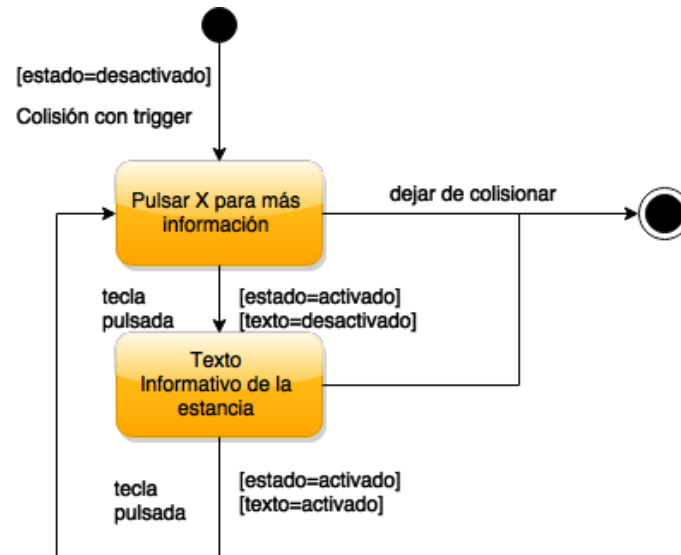


Ilustración 26. Diagrama de estado del gestor de ventanas emergentes.

5.4.7. Gestor de acceso a base de datos

Este módulo está compuesto por un solo script, DataBaseAccess. Este script sigue un patrón de diseño Singleton de acceso a base de datos en C#.

Este patrón está diseñado para limitar la creación de objetos correspondiente a una clase o el valor de un tipo a un único objeto. Consiste en asegurar que una clase tenga una instancia y también consiste en proporcionar un punto de acceso global a ella.

Al instanciar inicialmente este objeto, éste iniciará una conexión con la base de datos con un código similar al siguiente:

```

using System.Data.SqlClient;

SqlConnection myConnection = new SqlConnection("user id=username;" +
    "password=password;server=serverurl;" +
    "Trusted_Connection=yes;" +
    "database=database;" +
    "connection timeout=30");
    
```

Cada vez que se le solicitan textos correspondientes a códigos, o la lista de idiomas disponibles para ser mostrada en el menú de selección de idioma, se realizan consultas contra la base de datos a través de esta conexión.

5.5. Implementación de la aplicación del administrador

La implementación de la aplicación está integrada por tres módulos.

5.5.1. Vistas

Este módulo está compuesto por tres módulos, los cuales son: ViewLogin, ViewMenu, ViewInstance. Este agrupamiento de módulos se debe a que son la parte de la aplicación que interactuará con el administrador.

- **ViewLogin:** Esta página muestra al usuario el formulario que deberá rellenar para poder identificarse en la aplicación. En el momento que el usuario presione el botón de *iniciar sesión*, este módulo se comunicará con el controlador que gestiona las peticiones al servidor. De esta manera recibirá del controlador un mensaje de error en caso de que se haberse producido o se pasará a la siguiente vista, ViewMenu. Se encuentra codificado tanto en HTML como en PHP.
- **ViewMenu:** Esta vista hará peticiones a la base de datos, por medio del controlador, para obtener un listado de los enlaces de las estancias del templo. Cuando se presione uno de estos enlaces se enviará otra petición al servidor para cambiar de vista, ViewInstance.
- **ViewInstante:** Esta página tiene un formulario con los campos que se pueden modificar de la instancia que se ha seleccionado. Es aquí donde el administrador puede subir o eliminar los ficheros, cambiar o eliminar el idioma de la aplicación etc. Las peticiones que se hacen a la base de datos mediadas por el controlador.

5.5.2. RequestController

Realiza las peticiones a ServerController para obtener de la base de datos tanto la identificación del administrador, como para la lista de estancias y los campos que se desean añadir, actualizar o eliminar. Y una vez obtenida la respuesta la devuelve a la vista directamente. Puede decirse que tiene una función muy parecida a la del controlador del MVC.

5.5.3. ServerController

Este módulo contiene la lógica del programa, está compuesto por las consultas que se realizan a la base de datos. Los datos que devuelve este módulo tendrían una función parecida a la del modelo del MVC.

6. PRUEBAS

Este capítulo describe las pruebas realizadas para la aplicación de la visita, para ello se ha pedido a distintos usuarios que utilicen la aplicación.

Las pruebas desarrolladas han sido para el sistema. Estas pruebas verifican que la aplicación funciona correctamente en los entornos en los que es integrada.

A continuación se exponen las pruebas realizadas:

- **Pruebas de movimiento:** Se comprueba el desplazamiento y la rotación de la cámara en primera persona por el templo.
- **Pruebas de física:** Se comprueba, que la cámara no atravesase ninguna de las paredes del templo.
- **Pruebas de conexión con la base de datos:** Se comprueba que el idioma y los textos aparezcan en las paredes de las estancias que les corresponde.

6.1. Pruebas de movimiento

Este apartado verifica el correcto funcionamiento del movimiento de la cámara sobre el templo.

6.1.1. Prueba de movimiento de traslación

Se comprueba que las teclas presionadas y los desplazamientos coincidan con lo acordado por el requisito R18. Además de que la velocidad del desplazamiento se actualice adecuadamente cuando el usuario lo modifique.

6.1.2. Prueba de movimiento de rotación

Se comprueba que las teclas presionadas y el movimiento giratorio coincidan con el acordado en el requisito R19. También se comprueba que la velocidad de rotación se actualice correctamente cuando el usuario la modifique.

6.2. Pruebas de física

Se verifica que al desplazarse por el edificio histórico, el visitante no pueda atravesar las paredes. Uno de los fallos que se encontraron en esta prueba fue que el objeto

cámara al colisionar con una de las paredes del templo comenzaba a girar sin parar, esto se solucionó congelando el movimiento automático por defecto de la cámara.

6.3. Pruebas de conexión con la base de datos

Se comprueba que las peticiones hechas por la aplicación a la base de datos sean resueltas correctamente.

6.3.1. Prueba de aparición de textos

Para cada estancia se hace una comprobación de los textos correspondientes a cada pared.

6.3.2. Prueba de cambio de idioma

Se comprueba que el texto devuelto a la aplicación corresponde con el idioma seleccionado en el apartado de opciones del menú.

7. CONCLUSIONES Y TRABAJO FUTURO

Uno de los principales objetivos durante el desarrollo de este proyecto ha sido crear una visita guiada con fines didácticos, que permita el acceso a un monumento histórico para todos aquellos que no puedan visitar el templo de forma física.

Las aplicaciones implementadas han sido dos: el sistema de administración de textos, que permite introducir la información en la base de datos, y la aplicación de la visita que extrae la información del templo de la base de datos.

Tras el desarrollo de este proyecto se puede destacar:

- Se ha aprendido a gestionar el ciclo de vida del software.
- Se ha aprendido a modelar edificios en 3D con la herramienta de modelado 3D Blender.
- Se ha aprendido a retocar imágenes para crear texturas con las herramientas de tratamiento de imágenes Manga Studio 5 y GIMP 2.
- Se ha aprendido a integrar modelos 3D en herramientas de desarrollo como Unity.
- Se ha puesto en práctica el conocimiento para acceder a base de datos de servidores remotos.
- Se ha seguido una metodología de implementación modular en ambas aplicaciones.
- Se han realizado pruebas para verificar el correcto funcionamiento de la aplicación.

En conclusión este Trabajo de Fin de Grado ha logrado llevar a cabo con éxito todos los objetivos iniciales propuestos.

En relación a las líneas futuras de trabajo, se podrían realizar las siguientes mejoras:

- Obtener más fotos de las inscripciones jeroglíficas grabadas para tratarlas en las texturas de las paredes correspondientes y hacer más realista la visita.
- Añadir una nueva función didáctica que permita realizar preguntas de tipo test al usuario mientras realiza la visita.

8. REFERENCIAS

- [1] “Visitas virtuales de los Museos”, Ministerio de Educación, Cultura y Deporte área de cultura, <http://www.mecd.gob.es/cultura-mecd/areas-cultura/museos/visitas-virtuales.html>
- [2] “Patrimonio virtual”, España es cultura, http://www.xn--espaescultura-tnb.es/es/elementos_3D/
- [3] Dr. Bill Rosener, “3D Modeling Programs: Comparison of SketchUp and Blender”, <http://www.billrosener.com/teaching/publications/3D-modeling.pdf>, July 15 - 18, 2014
- [4] “Hacer el mapeo de UVs de una Malla (Mesh)”, Blender 2.6, http://wiki.blender.org/index.php/Doc:ES/2.6/Manual/Textures/Mapping/UV/Unwrap_ping
- [5] “Mapeado UV”, Blender 2.4, http://wiki.blender.org/index.php/Doc:ES/2.4/Manual/Textures/UV/Unwrapping_a_Mesh
- [6] “Renderizar”, Wikipedia, <https://es.wikipedia.org/wiki/Renderizaci%C3%B3n>
- [7] “Iluminación global”, Wikipedia, https://es.wikipedia.org/wiki/Iluminaci%C3%B3n_global
- [8] Matt Newman, “Beginners guide to accessing SQL Server through C#”, 22 Aug 2004, <http://www.codeproject.com/Articles/4416/Beginners-guide-to-accessing-SQL-Server-through-C>
- [9] “Visita virtual”, Templo de Debod memoria de Madrid, <http://templodedebod.memoriademadrid.es/visitaVirtual.html>
- [10] “Visita 3D”, Templo de Debod memoria de Madrid, <http://templodedebod.memoriademadrid.es/3d.html>
- [11] “Un pase por el Museo”, Museo del Romanticismo, <http://museoromanticismo.mcu.es/elMuseo/unPaseo.html>
- [12] “Museo de Antropología”, Museo de Antropología, <http://www.mcu.es/visitavirtualmuseos/museo-de-antropologia>
- [13] “Second Life”, Second Life, <http://secondlife.com/>
- [14] “Second Life”, Wikipedia, https://es.wikipedia.org/wiki/Second_Life
- [15] “¿Qué es Unity?”, Unity, <https://unity3d.com/es/unity>

- [16] “Welcome to the Unity Scripting Reference!”, Unity,
<http://docs.unity3d.com/ScriptReference/>

A. ANEXO: ADQUISICIÓN DE MATERIAL AUDIOVISUAL

Se visitó el Templo Egipcio de Debod para obtener fotos de las texturas que se iban a utilizar en el modelo 3D. Partiendo de la información proporcionada por el profesor Miguel Ángel Molinero Polo, se sacaron las fotos de las piedras originales del templo ya que una gran parte de las que lo componen son bloques de roca caliza creados para sustituir las piedras faltantes o deterioradas, entre estos elementos añadidos en su reconstrucción en Madrid se encuentra el techo.

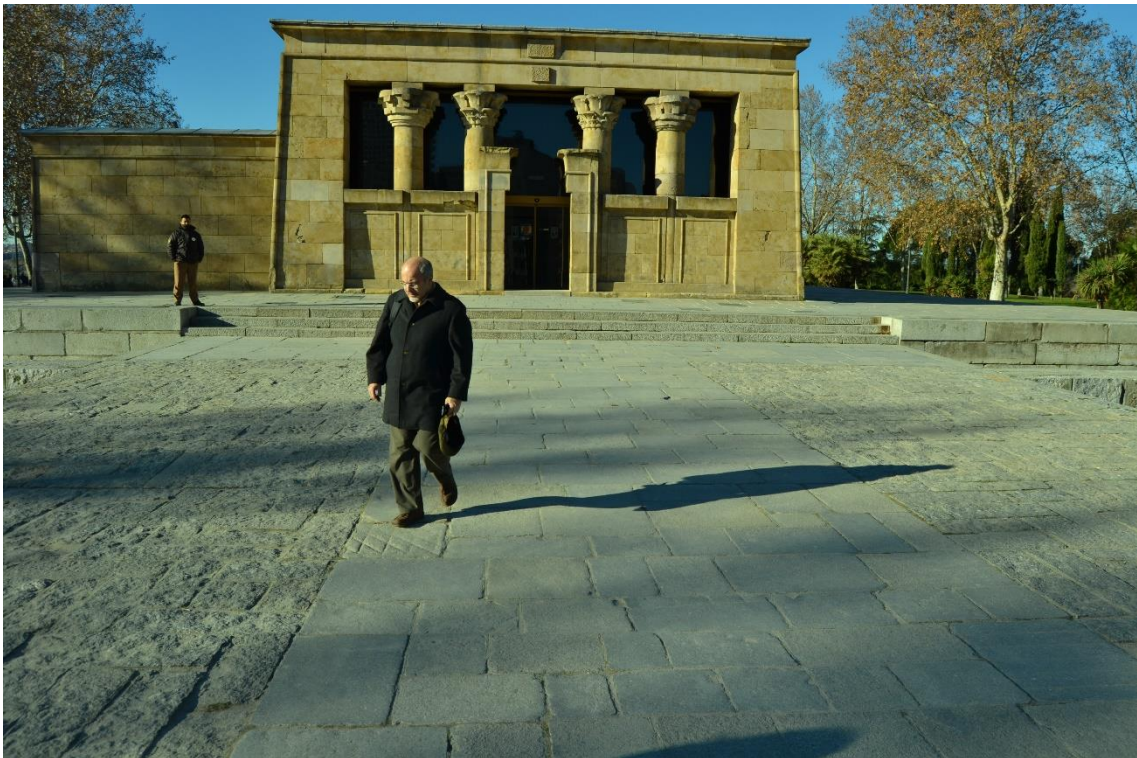


Ilustración 27 Visita al Templo Egipcio de Debod (profesor Miguel Ángel frente al Alzado Principal del templo).

B. ANEXO: CONCEPT ART

Después de la implementación del modelo 3D del templo y su posterior integración en la herramienta de desarrollo y aplicaciones interactivas Unity, se creó un concept art para tener definidos los estados a los que pasaría la aplicación con cada acción.

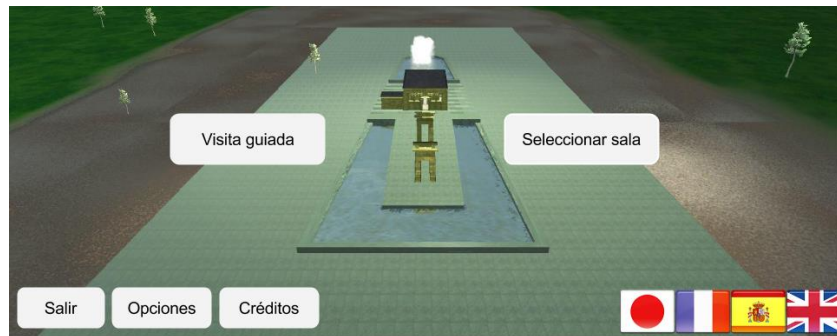


Ilustración 28. Menú principal.



Ilustración 29. Selección de salas.



Ilustración 30. Créditos de la aplicación.

Reconstrucción virtual de edificios históricos

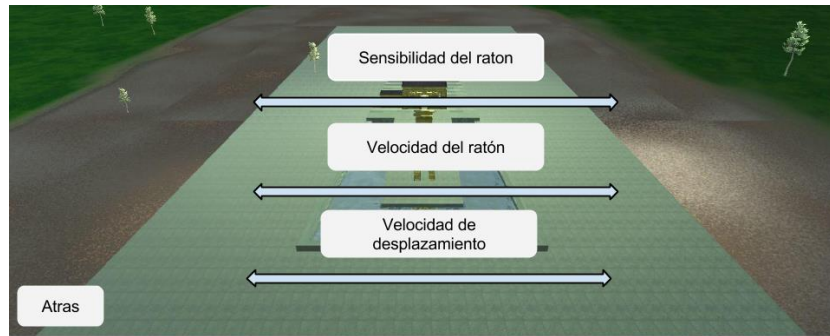


Ilustración 31. Opciones: configuración de las velocidades del desplazamiento y la rotación.

C. ANEXO: MANUAL DE USUARIO.

Al entrar en la aplicación de la visita 3D, estará en el menú principal. En él se puede cambiar el idioma dando clic en una de las banderas, ir a los créditos presionando el botón *Creditos*. Se puede cambiar la configuración de la velocidad del movimiento y la rotación seleccionando el botón *Opciones*.

La visita guiada irá estancia por estancia mostrando al visitante sus respectivos textos informativos hasta finalizar el recorrido. Para poder cambiar de estancias se ha de activar y desactivar los textos presionando la tecla X.

Si se elige la selección de salas, el visitante podrá realizar un recorrido libre en el que se moverá utilizando las siguientes teclas:

- **W:** Desplazamiento hacia adelante.
- **A:** Desplazamiento hacia la izquierda.
- **S:** Desplazamiento hacia atrás.
- **D:** Desplazamiento hacia la derecha.
- **Q:** Desplazamiento hacia abajo.
- **E:** Desplazamiento hacia arriba.
- **ArrowUp:** Giro de la cámara hacia arriba.
- **ArrowDown:** Giro de la cámara hacia abajo.
- **ArrowLeft:** Giro de la cámara hacia la izquierda.
- **ArrowRigth:** Giro de la cámara hacia la derecha.

El sistema de gestión de ficheros para el administrador se compone por una identificación, una lista de estancias de las cuales debe seleccionar la que se desea editar. En esta página de edición se podrán añadir ficheros a las respectivas paredes de la estancia.